Le NAS, ou **Network Attached Storage**, est un appareil de stockage autonome qui peut se connecter à votre réseau privé ou professionnel via Internet. Il permet de **sauvegarder, partager, sécuriser** mais aussi de **faciliter l'accès à vos fichiers** depuis plusieurs appareils.

Premièrement nous avons configuré notre machine virtuelle en y ajoutant trois nouveaux disques pour configurer avec un RAID 5. Pour cela, nous devons installer **mdadm**, c'est l'outil qui va nous permettre de configurer et gérer les matrices RAID. Mettre avant tout à jour la machine : **sudo apt update && sudo apt upgrade**.

Ensuite tapez la commande : **sudo apt install mdadm**.

On peut vérifier la dépendance avec la commande : **sudo mdadm -V**.

Identification des composants : **lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT**
Cela va nous permettre d'identifier les différents disques.

On va créer ensuite la matrice RAID 5, il faudra transmettre les composants avec la commande : **mdadm --create**, et spécifier le nom du périphérique à créer donc /dev/md0. Le niveau RAID et le nombre de périphériques : **sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd**.

Création et montage du système de fichiers sur la matrice :

- **sudo mkfs.ext4 -F /dev/md0**
- **sudo mkdir -p /mnt/md0** (pour attacher le nouveau système de fichiers).
- **sudo mount /dev/md0 /mnt/md0** (monter le système de fichiers).
- **df -h -x devtmpfs -x tmpfs** (vérifier si le nouvel espace est disponible).

Enregistrement de la disposition du tableau, pour assurer que le tableau soit réassemblé automatiquement au démarrage, il faut ajuster le fichier **/etc/mdadm/mdadm.conf**.
Commande pour vérifier l'assemblage de la matrice : **cat /proc/mdstat**.

Analyser automatiquement le tableau actif et ajouter le fichier en saisissant : **sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf**.

Mise à jour de l'initramfs, ou système de fichiers RAM initial (afin que la matrice soit disponible) : **sudo update-initramfs -u**

Ajout des nouvelles options de montage du système de fichiers au **/etc/fstab**, pour le montage automatique au démarrage : **echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab**.

Création des utilisateurs avec samba, premièrement on met à jour avec la commande **apt update**, puis on installe les packages Samba nécessaires avec leurs dépendances : **apt install samba**.

On crée et ajoute les utilisateurs dans le groupe samba : **sudo smbpasswd -a *nomutilisateur***.

Il faudra créer les partages auxquels ils auront accès : **mkdir *nomdudossier***.

Modifier le fichier de configuration Samba : **sudo nano /etc/samba/smb.conf**.
Il nous permettra d'autoriser l'accès aux dossiers et configurer d'autres paramètres.
À l'intérieur on y entre les commandes suivantes en fonction de l'utilisateur utilisé et du dossier créer :
[nomdudossier]
path = nomdudossier
read only = no
guest ok = no
valid users = ...

Pour finir il faudra redémarrer Samba avec la commande : **sudo systemctl restart smbd.service**.
Attention: Si vous avez créé votre dossier en suivant un chemin, il faudra mettre le chemin sur **path.**

On pourra ensuite vérifier l'accès aux dossiers partagés depuis le serveur lui-même en installant le service client avec : **sudo apt install smbclient -y**.
Puis taper la commande : **smbclient -U *nomutilisateur* //[IP_address|Server_name]/*nomdudossier* -c 'ls'**

Pour déplacer les utilisateurs dans le RAID, nous devons installer avant tout le **rsync** en mode root, avec la commande : **sudo apt-get install rsync**

Ensuite avec la commande **rsync -av /chemindudossier /mnt/md0** l'utilisateur sera déplacé dans le RAID.

Sécuriser avec SFTP, installer les packages après avoir mis à jour la machine : **apt update**
apt install proftpd -y

Vérifier que le serveur est bien actif : **sudo systemctl status proftpd**

Ensuite pour configurer le pare-feu on va devoir télécharger UFW : **apt install ufw -y**
Activer UFW : **sudo ufw enable**
Là nous avons un réseau sécurisé, mais pour encore plus le sécuriser, on peut le mettre dans un port spécifique. Nous avons choisi le port **6500**.
Dans le fichier **/etc/ssh/sshd_config**, on entre **Port 6500** et pour le pare-feu on tape la commande : **sudo ufw allow 6500/tcp**

Penser à restart (**sudo systemctl restart proftpd** && **sudo systemctl restart ssh**), puis faire le test avec : **sftp -P 6500 *nomutilisateur@IP_Adress***

Configurer SFTP

Créer des utilisateurs SFTP :
sudo adduser *nomutilisateur*
sudo passwd *nomutilisateur*

Dans le fichier **/etc/ssh/sshd_config**, il faut autoriser les utilisateurs à se connecter au réseau avec: **AllowUsers *nomutilisateur***
Il faut ensuite créer un répertoire dédié dans le RAID pour la sauvegarde avec les commandes suivantes :
sudo mkdir -p /mnt/md0/sftp
sudo chown root:root /mnt/md0/sftp
sudo chmod 755 /mnt/md0/sftp
sudo mkdir -p /mnt/md0/sftp/*nomutilisateur*
sudo chown *nomutilisateur*:*nomutilisateur* /mnt/md0/sftp/*nomutilisateur*

Nous avons rencontré un problème au niveau du SSH (la connexion en ssh n'était plus possible d'un coup), les commandes suivantes ont permis de résoudre ce problème :
sudo ufw allow OpenSSH (Permet d'autoriser SSH sur le port 22)
sudo ufw enable (active le pare-feu)
sudo ufw status (permet de voir le status)

WebDAV

Tout d'abord il faut mettre à jour les paquets. Ensuite installer Apache2 avec : **sudo apt install apache2**
Puis activer le module WebDAV :

sudo a2dissite 000-default
sudo service apache2 reload

Ensuite il faut configurer un hôte virtuel. Créer un fichier de configuration avec : **sudo nano /etc/apache2/sites-available/webdav.local.conf** et ajouter le contenu suivant:

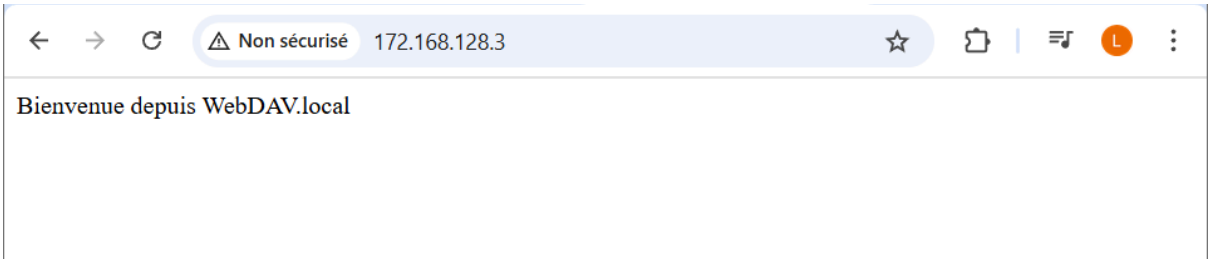
<VirtualHost *:80>
ServerAdmin webmaster@localhost
Servername webdav.local

DocumentRoot /var/www/webdav
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
<Directory /var/www/webdav/>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>
</VirtualHost>

Nous avons créer le répertoire /var/www/webdav/, le fichier d'index de test et défini la propriété correcte en tapant les commandes:
sudo mkdir /var/www/webdav
sudo sh -c 'echo "Bienvenue depuis WebDAV.local" > /var/www/webdav/index.html'
sudo chown www-data:www-data /var/www/webdav

Activer le nouveau site et recharger Apache2 avec :
sudo a2ensite webdav.local
sudo service apache2 reload

Après ces configurations, nous pouvons déjà tester en naviguant vers **http://[IP_Serveur]** et nous voyons bien « **Bienvenue de WebDAV.local** ».

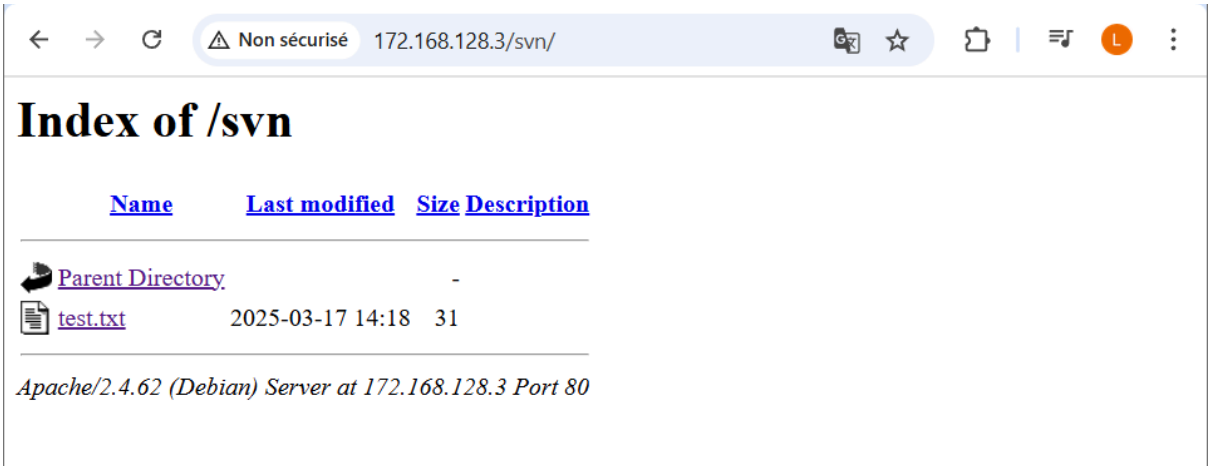


Cette page confirme que le serveur Apache est opérationnel.

À présent, il faut activer le module WebDAV avec : **sudo a2enmod dav_fs**
Puis créer un répertoire pour les données WebDAV et un fichier de données arbitraire à des fin de test:
sudo mkdir /var/www/webdav/svn
sudo touch /var/www/webdav/svn/linuxconfig.txt
sudo chown www-data:www-data /var/www/webdav/svn

Modifiez la configuration de l'hôte virtuel **/etc/apache2/sites-available/webdav.local.conf** pour inclure les paramètres WebDAV en ajoutant :
Alias /svn /var/www/webdav/svn
<Location /svn>
DAV On
</Location>
Toujours important de redémarrer le système : **sudo service apache2 restart**

Test en navigant vers [http://[IP_Serveur]/svn](http://[IP_Serveur]/svn).



Pour configurer l'authentification utilisateur, on crée un fichier de mot de passe et ajoute un utilisateur:
sudo mkdir /usr/local/apache2/
sudo htpasswd -c /usr/local/apache2/webdav.passwords *nomutilisateur*

Et aussi modifier l'hôte virtuel; en ajoutant:

<Location /svn>
DAV On
AuthType Basic
AuthName "webdav"
AuthUserFile /usr/local/apache2/webdav.passwords
Require valid-user
</Location>

On redémarre et on fait le test.